GRASS ENGINE

Plantage

Object Tab

All following objects are absolutely compatible to each other. The plantage object is based upon the well-known HairDepartment (HD) technology by BGS-Group (www.bgs-group.de).

DPIT R3 takes this technology to a much higher level. So you might know a lot of the Plantage settings already.

Plantage requires an object with a surface as a first child. The grass/ hair will then be generated randomly on the surface of this object.

Render only : Checkbox
Will only generate the grass when you render your scene.

Load H.Dep. settings : Button
If you have old, saved HD settings, you can directly load them into Plantage by clicking on the button.

Non-destructive : Checkbox
Will not remove the surface object

Random : Integer
Seed value for the randomness

Mode : Drop down

Splines	Generates Splines
Polygons	Obsolete and should not be used any longer

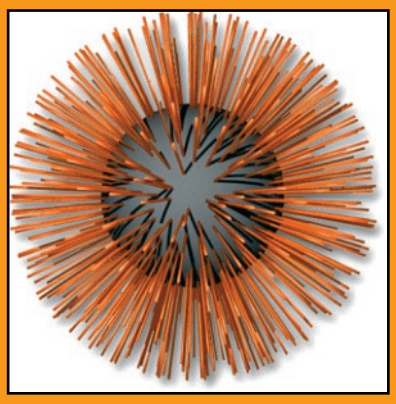
Advice: You can then use the *PlantVolume* object with Plantage. The reason is that splines are generated much faster than polygons and are displayed faster than polygons in the editor view.

Interpolation : Drop down
Intermediate points : Integer
See the Cinema 4D user manual -> Splines

View type : Drop down
(see Transplant object for more information)

Unlimited	All blades are generated
Roots in view	Only the blades where the roots are visible
All in view	This will take all points in the view into account

All other options in the Object Tab are identical to the options in the Transplant object. Please refer to these to identify the functionality.



Density Tab

Set Object settings : Button
changes some settings in relation to the size of the currently used surface object.

All other options in the density tab are equal to the ones of the density tab in the Transplant object. Please refer to the Transplant descriptions.

Appearance Tab

Distance Variation : Percentage slider
This varies the distance and direction between the spline segments

Length : Float
The maximum length of the spline blades

Length min. : Float
Minimum length for the length map

Length Map : Link
Here you can drag a vertexmap into, that works as a length map. A weight of 100% means full length and 0% means minimum length.

Shader : Link
A shader can be used to define the length (luminance, UV Space)

Length variation : Percentage
Varies the length in positive and negative direction

Sections Editor/Render : Integer slider
The subdivision of the spline. Spline point count = Segments+1

Alignment Tab

Gravity : Float
Defines the strength of gravity

Direction X/Y/Z : Float
Defines the gravity direction

Angle variation : Percentage
Varies the angle of the whole blade

...to vertical
Most of the time not used.

PlantVolume

The PlantVolume is especially used in conjunction with Plantage. It generates the polygonal blades. This is much faster than Plantage generating the polygonal blades as it would need to regenerate the splines each time a setting changed.

It therefore requires a child object as input that generates splines (Plantage, SplineGen...).

You can also use a second, diameter spline as second child of the PlantVolume object that defines the diameter shape.

Render Only : Checkbox
Forces the object to generate the polygons only for rendering

Radius : Float
Defines the radius of the blades

Sidefaces : Integer
Specifies the sidefaces count

Banking : Checkbox
Banks the rotation of the polygonal hull

Link Depth to Width : Checkbox
Links the depth spline graph (Shape W) to the width spline graph (Shape H)

Shape W/H : Spline graph
Here you can define the shape along the splines of the polygonal hull.

Auto : Checkbox
Automatically uses the full animation range, so all frames of your scene for the caching

Start/Stop : Time
The start and end for the caching can be set in here. Defined in frames

Cache Mode : Checkbox
Uses the cached solution (if you have created it already)

Calculate Cache : Button

Empty Cache : Button
Removes the cached solution

Polygon animation : Checkbox
Should only be used when the polygonal structure changes during animtion. For example when the spline points change the polygonal hull will change. This uses much more memory!!

PlantWind

PlantWind adds windeffects to any spline! One segment or multi segment splines, with or without interpolation. This is a huge advantage in contrast to the HD animation object, for you could also use it with DPIT splines.

Another addition is that you can restrict the wind effect to a certain cuboid or ellipsoid area. Simply set **Restriction** to **Cuboid** or **Ellipsoid** instead of **Unlimited**. If you do so, the **Size** vector field is enabled and you can set the dimensions of the restriction field.

The **Strength** of the Wind effect can be set in the according field with the same name.

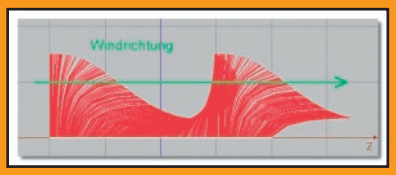
If you have chosen Ellipsoid/Cuboid (E/C) as restriction you also have the chance to use a spline **Falloff** to control the strength. The right side of the spline signifies the center of the E/C (or if you are using the **Minimum** value, then the right side of the spline signifies the surface of the minimum E/C) and the left side the surface of the E/C.

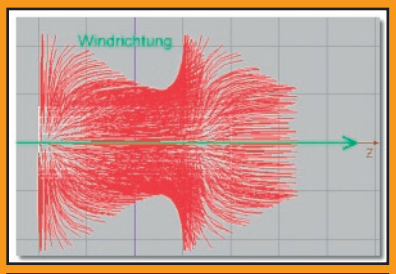
This way for example you could let a helicopter fly through a grassfield and only the grass near the helicopter will move stronger due to the wind the rotor blades produce.

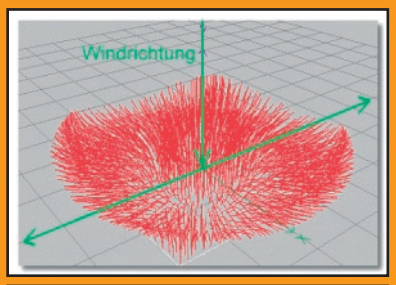
The *Minimum* value will span another ellipsoid inside of the main ellipsoid. Everything inside this minimum ellipsoid will receive the strength of the left side in the spline control.

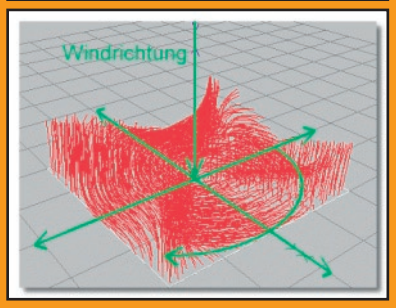
The **Velocity (m/s)** parameter defines at which speed the waves defined by the **Wave length** (the length of the wave moving along the grass) should move along the surface object. The speed is directly linked to the Wave length and the duration of the animation. This might be important when animating loops.

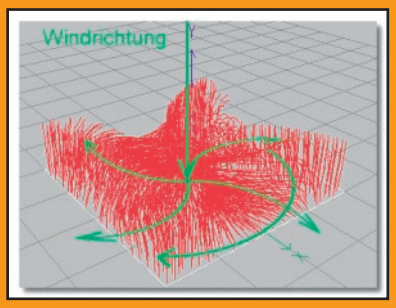
If you want to create a wave with a length of 200 units, which should restart seamlessly after the end of a 1 second animation cycle, the speed value has to be the same as the wave length (200).
If the animation cycle has a duration of 2 seconds, you need a speed value of 50% of the wave length. Using a value of 100% would cause the wave motion to repeat two times within the animation cycle.

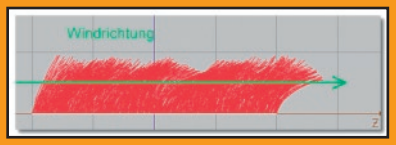












There are 6 **Function**s that you can apply to the grass and that will animate it. This is the core of PlantWind, so read it carefully.

The **Standard** function type oscillates the hairs using a mathematical function and creating linear
waves on a flat plane. The direction of the hairs is defined by the horizontal plane of the wind direction.

The **Axial** function type is similar to the Standard function type and creates linear waves along the suface
object. The difference is that the direction of the grass is defined by the axis of the wind direction.

The **Radial** function type creates concentric waves along the grass. The direction
of the grass blades is defined by the plane vertical to the wind direction. A pitch value of
90 Degree on a vertical surface plane would generate a result as shown in the example beside.

The **Circular** function type creates radial waves which move around a central point. The direction of the
grass is defined by the plane vertical to the wind direction. A pitch value of 90 degree on a vertical
surface plane would generate a result as shown in the example beside.

The **Twister** function type creates waves similar to the circular one, which however rotate around a central point
(like a whirl pool or „eddy"). The direction the grass is defined by the plane vertical to the wind direction.

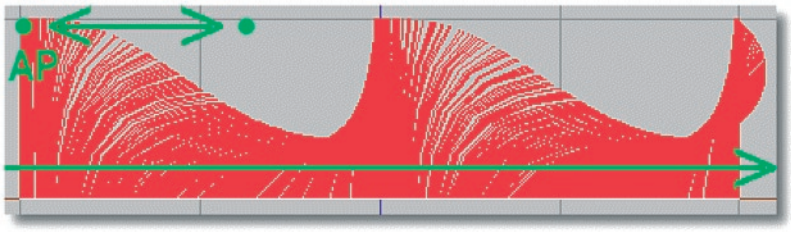
Note: As it is very difficult to determine the wave length on grass planes with Circular or Twister functions, the following rule can be used to approximate the result:
*Wavelength = strength * 2PI / wave count (2PI = 6,283)*
Most of the time guessing a good value is the fastest way though.

The **Noise** function type creates a random wave field along the grass. The direction of the grass is defined by the horizontal plane
of the wind direction.

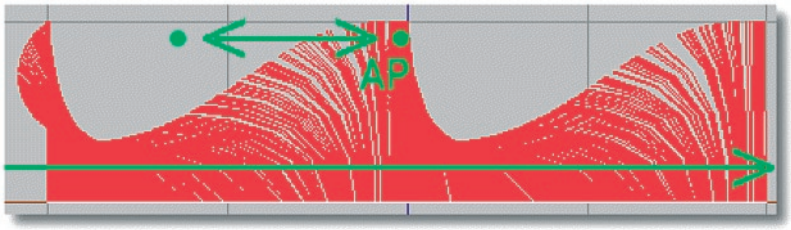
PlantWind will additionally display the certain directions for you in the editor view!

All of the main functions do have 11 **Subtype**s (Subfunctions) that allow even more variation. The subtypes are indexed from 0 to 10.

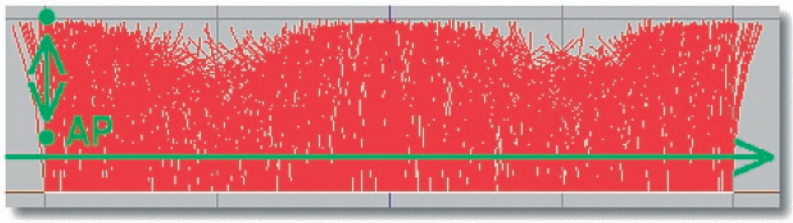
00: The grass swings from side to side beginning at the starting position (**SP**) and along with the wind direction (**WD**).



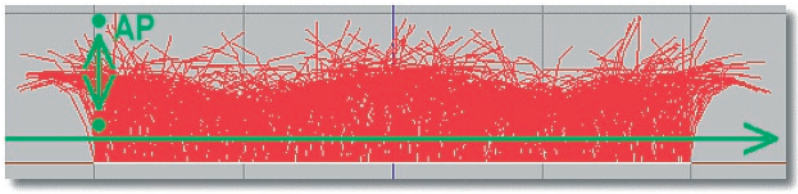
01: The grass swings from side to side beginning at the **SP** and against the **WD**.



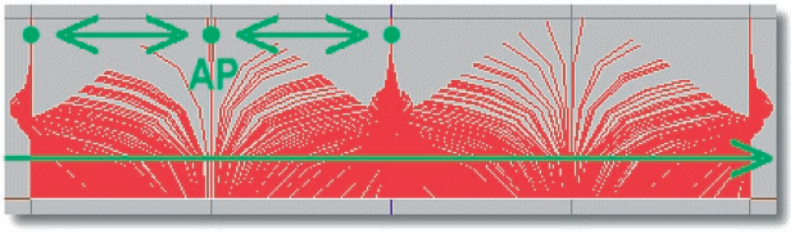
02: The grass is pulled up along from the **SP** and along the wind directions **WD**. This effect is best used with grass that has been created using gravity.



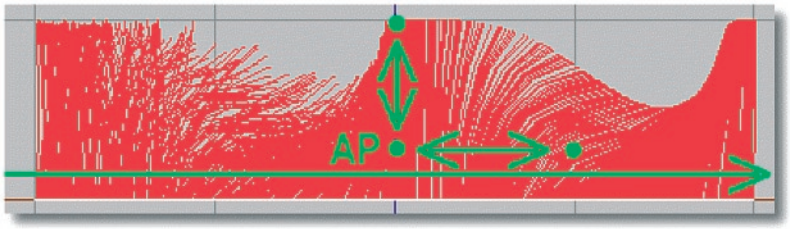
03: The grass is pushed down along from the **SP** and along the **WD**. This effect is best used with grass that has been created using gravity.



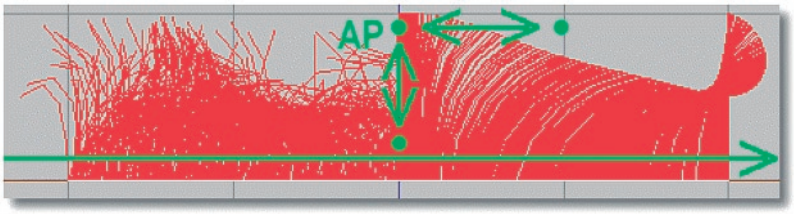
04: The grass swings from the **SP** along and against the **WD**.



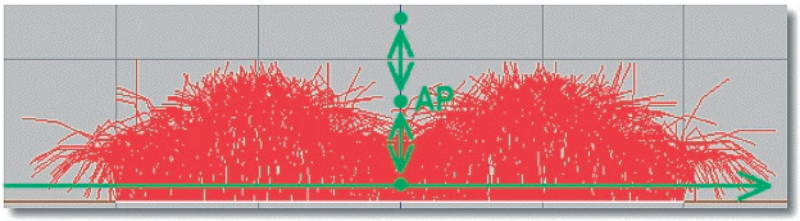
05: The grass swings from side to side beginning at the **SP** and along the **WD** and are also pulled up. This effect is best used with grass that has been created using gravity.



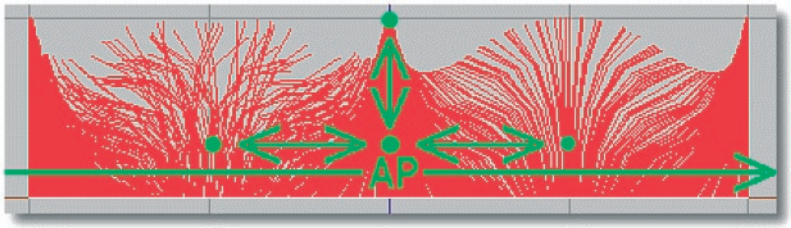
06: The grass swings from side to side beginning at the **SP** and along the **WD** and are also pulled down. This effect is best used with grass that has been created using gravity.



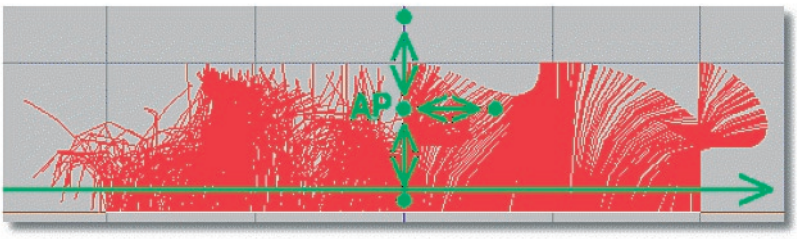
07: The grass is pulled up and down beginning at the **SP** and along the **WD**. This effect is best used with grass that has been created using gravity.



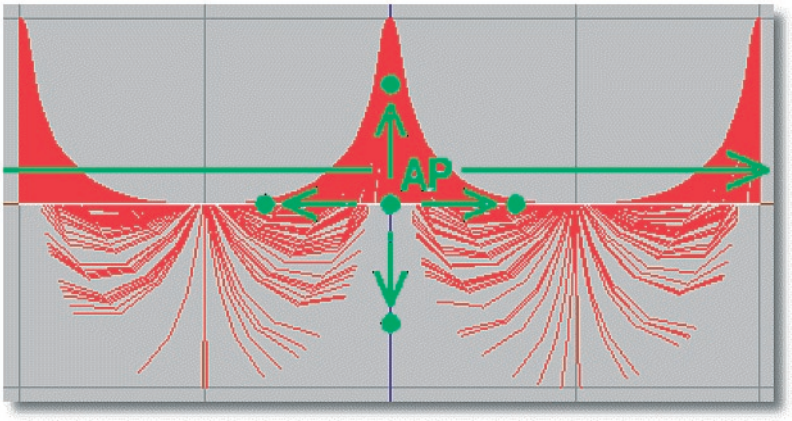
08: The grass swings along and against the **WD** beginning at **SP** and are also pulled up. This effect can be used both on hairs which have been created with or without gravity.



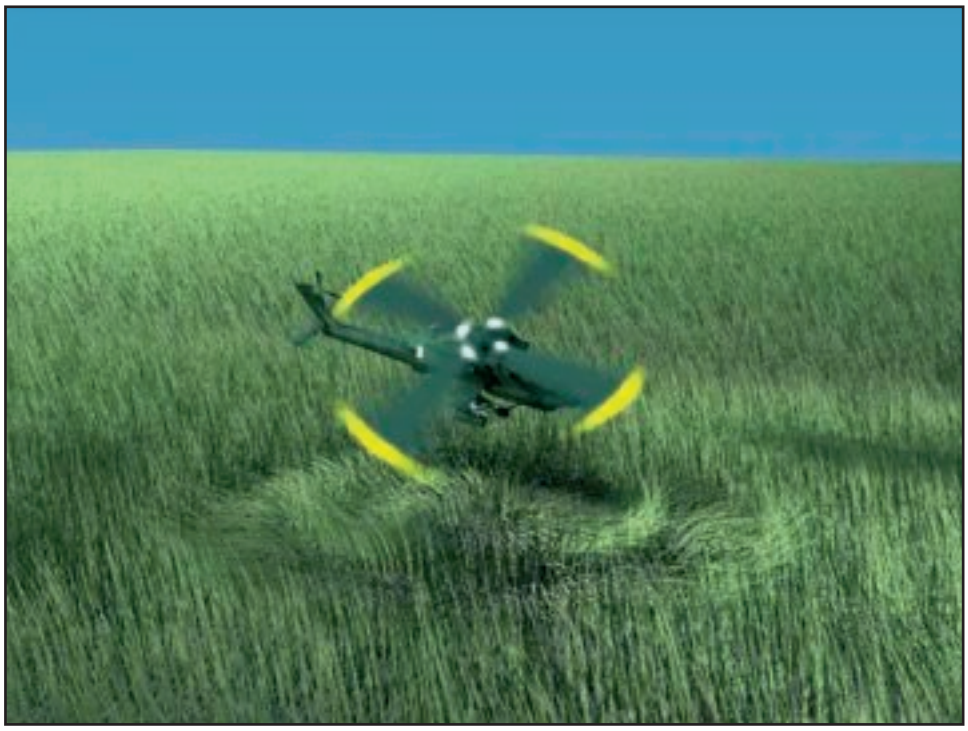
09: The grass swings along and against the **WD** beginning at **SP** and are also pulled up and down.



10: The grass is first pulled down beginning at the **SP** along the **WD**, then against the **WD** and then back up again.



The **Affect** option defines which parts are affected when animating. **All** will affect all points along the spline, whereas **Root** will affect the angle of the spline itself.



With the **Offset** value you can shift the motion of the wave within the animation. The entered value equals the number of seconds in the animation.
Hair Solidity can be used to define the stability of the grass when it´s affected by the wind. The higher you set this value the less the grass will bend in the wend.

Last but not least, you can add an additional noise wave to the grass motion. To activate it check the **Use extra noise** option.

The following value, **Noise strength**, defines the strength of the noise in percent.

Example: If the main wave has a strength of 80, the value 0.250 (25%) will lead to a strength of 20.
The next value to be set, **Noise length**, is the length of the noise wave. This value also depends percentually on the length of the main wave.

PlantWind can be used on splines and PlantVolume polygons!

PlantLength

PlantLength´s intention is to let you grow your grass (or hair) splines. But of course you can use it on any other Cinema 4D spline!

Length types : Drop down
There are 2 different types:
Point-by-Point & **Whole Spline**

Point-by-Point will grow the spline by minimizing the distance between the currently processed point and the point before. If this method is for example used on a DPIT R2 pathspline, then the shape of the DPIT object will not move with the length.

The **Whole spline** method though will move the points (in the segment) toward the first point (of the segment). The result is then similar to the behavior of R2´s own growth parameter. The shape will move with the length.

Invert : Checkbox
Inverts the growth, so it grows from the last point to the first point

PlantLength is a deformer and must be inserted under the spline that you want to grow.

PlantCollision

Collision type : Drop down
Ellipsoid Ellipsoid Falloff shape
Cuboid Cuboid Falloff shape

Collision detection on splines. All splines that are within the Ellipsoid/ Cuboid (E/C) area are moved outside of it.

Size : Vector
This sets the dimensions of the Ellipsoid/Cuboid.

You also have the chance to use a spline **Falloff** to control the strength. The **left** side of the spline signifies the center of the E/C (or if you are using the **Minimum** value, then the left side of the spline signifies the surface of the minimum E/C) and the right side the surface of the E/C.

Keep Length : Checkbox
tries to keep the length of the each spline segment constant.

Iterations : Integer slider
Is a big helper concerning collision detection. This value will force PlantCollision to go n-times through the collision detection. This is sometimes necessary if you are using Keep Length, because Keep Length is evaluated after the collision detection. So Iterations might reduce artifacts.

Usually an iteration value between 2 and 7 is more than enough.

Object : Link
The position of the E/C is by default at the deformers position. If you want it to move with another object, simply drag an object from your Object Manager into the link field.

Note: The ellipsoid collision detection should be prefered as the algorithm is evaluated much better than the Cuboid one.

Floor collision : Checkbox
Another (but simple) collision detection. This will use an object´s global Y-position for the collision detection. If a spline point´s global Y-position lies under this value it´s set back to the object´s global Y-position.

Floor object : Link
The Floor object must be dragged into the link field with the same name, to activate the calculation.

Note: Currently there is no full polygon collision detection.